

Exercise work 3

Name: Pekka Lehtola

Do you need help/comments:

Koodi paikoittain kommentoimatta, koska aika loppui kesken (Firman projekti vei viikosta suuren osan ajasta...)

Schedule:

- Week 1: Core game mechanics working.
- Week 2: UI, UML diagrams, Flow chart
- Week 3: Random events, Inheritance, Polymorphism. Ui class separated to multiple classes
- Final: Updated graphics, UML diagrams and Flow chart final version.

Challenge goal:

Challenging (Grade 4-5)

Work done this week: 8h

What has been implemented / changed:

Ui class separated to multiple modules / classes.

Inheritance from Ui class to other Ui elements.

Random events as power ups that appear into the level.

Ui elements:

```
import pygame
from config import *
from ui import Ui

class Heart_ui(Ui):

    def __init__(self):
        super(Heart_ui, self).__init__("images/hearth_1.png")
```

Heart Ui otherwise unmodified.

```
import pygame
from config import *
import time
from ui import Ui

class Ammo_ui(Ui):
    def __init__(self, image):

        super(Ammo_ui, self).__init__(image)

        # Used for characters reload animation.
        self.reload_list = ["images/circle1.png", "images/circle2.png", "images/circle3.png", "images/circle4.png"]
        self.counter = 0
```

Modifications to show_ammo function.

```
# Shows every ammo that character has in ammo list
def show_ammo(self, character, screen):

    y = 10

    if character.name == "Cowboy":

        x = 60

        for ammo in character.ammo:

            if ammo.super == True:

                ammo.rect.x = x
                ammo.rect.y = 90
                screen.blit(ammo.surf, ammo.rect)
            else:
                ammo.rect.y = y
                ammo.rect.x = x
                screen.blit(ammo.surf, ammo.rect)

                x += 30
```

Ammo class:

```
main.py x ammo.py x heart_ui.py x ammo_ui.py x heart_power_up.py x character.py x ui.py x config.py x sup
1  # File name: character
2  # Author: Pekka Lehtola
3  # Description: Ammo class for characters to shoot.
4
5  import pygame
6  from config import *
7
8  # Ammo is derived from pygame sprites
9  class Ammo(pygame.sprite.Sprite):
10
11     def __init__(self, ammo, name):
12         super(Ammo, self).__init__()
13         self.name = name
14         self.surf = pygame.image.load(ammo) # Image
15         self.surf.set_colorkey(GREEN_SCREEN) # See through color
16         self.rect = self.surf.get_rect() # Hitbox
17
18         self.shot = False
19
20         self.super = False
21
22     # Function for shot ammo
23     def ammo_shot(self, shooter, enemy):
24
25         if self.shot == True:
26
27             # If the shooter is indian arrow travels to left
28             # If arrow hits cowboy, cowboy is killed.
29             if shooter.name == "Indian":
30                 self.rect.move_ip(-BULLET_VELOCITY, 0)
31
32                 if self.rect.colliderect(enemy.rect):
33                     self.rect.x = -2200
34                     self.kill()
35
36                     if self.super:
37                         for heart in range(3):
38
39                             if len(enemy.health) == 0:
40                                 enemy.kill()
41
42                             else:
43                                 enemy.health.pop(-1)
44
45                     else:
46
47                         if len(enemy.health) == 0:
48                             enemy.kill()
49
50                         else:
51                             enemy.health.pop(-1)
52
53             # Same as indian but bullet travels right
54             if shooter.name == "Cowboy":
55                 self.rect.move_ip(BULLET_VELOCITY, 0)
56
57                 if self.rect.colliderect(enemy.rect):
58                     self.rect.x = 2200
59                     self.kill()
60
61                 if self.super:
62
63                     for heart in range(3):
64
65                         if len(enemy.health) == 0:
66                             enemy.kill()
67
68                         else:
69                             enemy.health.pop(-1)
70
71                 else:
72
73                     if len(enemy.health) == 0:
74                         enemy.kill()
75
76                     else:
77                         enemy.health.pop(-1)
```

Ui class:

```
main.py x ammo.py x heart_ui.py x ammo_ui.py x heart_power_up.py x character.py x ui.py x config.py x sup
1  # File name: ui
2  # Author: Pekka Lehtola
3  # Description: User interface class. Used for showing different elements in screen.
4
5  import pygame
6  from config import *
7  from time import sleep
8  #from main import main
9
10
11 class Ui(pygame.sprite.Sprite):
12
13     def __init__(self, image):
14         super(Ui, self).__init__()
15         self.surf = pygame.image.load(image)
16         self.surf.set_colorkey(GREEN_SCREEN)
17         self.rect = self.surf.get_rect()
18
19     ### WORK IN PROGRESS ###
20
21     # In the start of the game shows count down
22     def show_element(self, screen):
23
24
25         self.rect.centerx = (SCREEN_SIZE_HOR / 2)
26         self.rect.centery = (SCREEN_SIZE_VER / 2)
27         screen.blit(self.surf, self.rect)
28         pygame.display.flip()
29         sleep(1.2)
30
31     # Will be used to show the winner.
32     #def game_end(self):
33
34     def countdown(self, screen):
35
36         self.surf = pygame.image.load("images/33.png")
37         self.surf.set_colorkey(GREEN_SCREEN)
38         self.show_element(screen)
39         screen.blit(BG, (0, 0))
40
41         self.surf = pygame.image.load("images/22.png")
42         self.surf.set_colorkey(GREEN_SCREEN)
43         self.show_element(screen)
44         screen.blit(BG, (0, 0))
45
46         self.surf = pygame.image.load("images/11.png")
47         self.surf.set_colorkey(GREEN_SCREEN)
48         self.show_element(screen)
49         screen.blit(BG, (0, 0))
50
51     def winner(self, cowboy, indian, screen):
52
53
54         if len(indian.health) == 0:
55
56             self.surf = pygame.image.load("images/cowboy_wins.png")
57             self.surf.set_colorkey(GREEN_SCREEN)
58             self.show_element(screen)
59             sleep(1)
60             return True
61
62         if len(cowboy.health) == 0:
63
64             self.surf = pygame.image.load("images/indian_wins.png")
65             self.surf.set_colorkey(GREEN_SCREEN)
66             self.show_element(screen)
67             sleep(1)
68             return True
69
70
```

Main 1:

```
main.py × ammo.py × heart_ui.py × ammo_ui.py × heart_power_up.py × character.py × ui.py × config.py × sup
1  # File name: main
2  # Author: Pekka Lehtola
3  # Description: main file exercise work
4
5  import pygame
6  from config import *
7  from character import *
8  from ui import *
9  from time import perf_counter, sleep
10 from heart_ui import *
11 from ammo_ui import *
12 from heart_power_up import *
13 from super_ammo import Ammo_power_up
14 from random import randint
15
16 running = True
17
18
19
20 def main():
21
22     # Initialize pygame
23     pygame.init()
24
25     # Sets up screen size
26     screen = pygame.display.set_mode([SCREEN_SIZE_HOR, SCREEN_SIZE_VER])
27
28     heart_timer = randint(7, 12)
29     super_ammo_timer = randint(14, 20)
30     global running
31
32     # Creates indian and cowboy objects.
33     # Also define graphics to them.
34     indian = Character("images/indian_1.png", "Indian")
35     cowboy = Character("images/cowboy_1.png", "Cowboy")
36
37     # Adds indian and cowboy to sprite group.
38     all_sprites = pygame.sprite.Group()
39     all_sprites.add(indian)
40     all_sprites.add(cowboy)
41
42     # Indians initial location
43     indian.rect.x = 1820
44     indian.rect.y = 640
45
46     # Cowboys initial location
47     cowboy.rect.x = 100
48     cowboy.rect.y = 640
49
50     heart = Heart_ui()
51     reload = Ammo_ui("images/circle1.png")
52
53     heart_power_up_list = []
54     super_ammo_power_up_list = []
55
56     heart.set_up_hearts(character=cowboy)
57     heart.set_up_hearts(character=indian)
58
59     # Background update
60     screen.blit(BG, (0, 0))
61
62     # start countdown
63     countdown = Ui("images/11.png")
64     countdown.countdown(screen)
65
66     winner = Ui("images/cowboy_wins.png")
```

Main 2:

```

66 winner = Ui("images/cowboy_wins.png")
67
68 while running:
69
70     timer = perf_counter()
71
72
73     # Detects events while game is running.
74     for event in pygame.event.get():
75         if event.type == pygame.QUIT:
76             running = False
77
78     # Pygame detects what keys are pressed.
79     pressed_keys = pygame.key.get_pressed()
80
81     # The term used for rendering objects is blitting.
82     # Surf : Surface, Rect : Rectangle.
83     # Rectangle is used for collide detection
84     # Surface can be image or color.
85
86     for characters in all_sprites:
87         screen.blit(characters.surf, characters.rect)
88
89     # Ui elements
90     heart.show_hearts(cowboy, screen)
91     heart.show_hearts(indian, screen)
92
93     reload.show_ammo(cowboy, screen)
94     reload.show_ammo(indian, screen)
95
96     reload.show_reload(indian, screen)
97     reload.show_reload(cowboy, screen)
98
99
100    # Reload function
101    indian.reload(pressed_keys, timer)
102    cowboy.reload(pressed_keys, timer)
103
104    # Shooting function
105    indian.shoot(pressed_keys, timer)
106    cowboy.shoot(pressed_keys, timer)
107
108    # Moving function
109    indian.move(pressed_keys)
110    cowboy.move(pressed_keys)
111
112    # Bullet travel and hit detection
113    for ammo in indian.shot_ammo:
114
115        ammo.ammo_shot(indian, cowboy)
116        screen.blit(ammo.surf, ammo.rect)
117
118    for ammo in cowboy.shot_ammo:
119
120        ammo.ammo_shot(cowboy, indian)
121        screen.blit(ammo.surf, ammo.rect)
122

```

Main 3:

```
123 # Power ups
124
125 if timer > heart_timer:
126     heart_timer = timer + randint(7, 12)
127     heart_power_up = Heart_power_up()
128     heart_power_up.add_power_up(heart_power_up_list)
129
130
131 for hearts in heart_power_up_list:
132     hearts.show_power_up(screen)
133     hearts.heart_pickup(indian, cowboy, heart)
134
135 if timer > super_ammo_timer:
136     super_ammo_timer = timer + randint(14, 20)
137     super_ammo = Ammo_power_up()
138     super_ammo.add_power_up(super_ammo_power_up_list)
139
140
141 for super_ammo in super_ammo_power_up_list:
142     super_ammo.show_power_up(screen)
143     super_ammo.super_pickup(indian, cowboy)
144
145
146
147
148
149 # Update screen
150 pygame.display.flip()
151
152 # Background update
153 screen.blit(BG, (0,0))
154
155 # Starts game over when either character dies.
156
157 if winner.winner(cowboy, indian, screen):
158     main()
159
160 pygame.quit()
161
162 main()
```

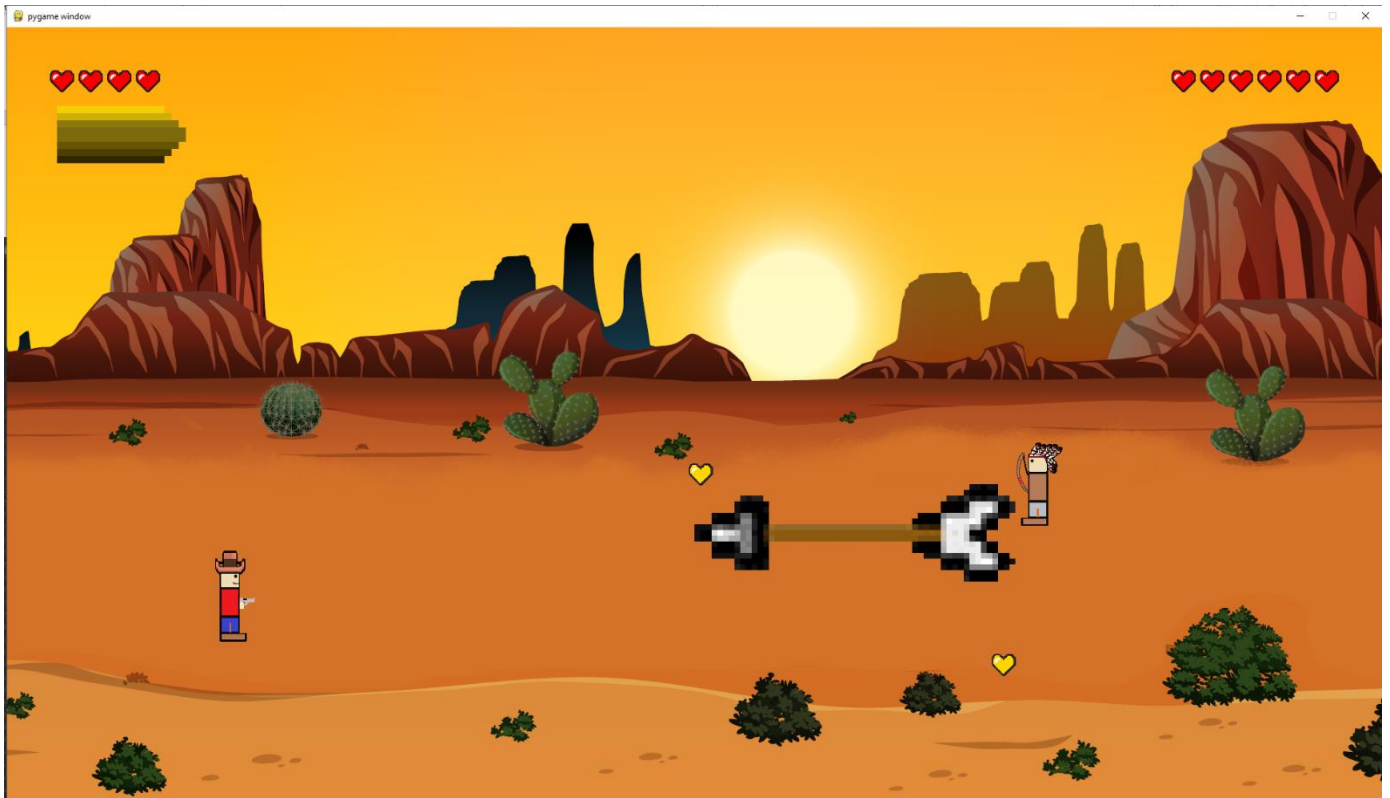
Heart power up class:

```
main.py × ammo.py × heart_ui.py × ammo_ui.py × heart_power_up.py × character.py × ui.py × config.py × sup
1  # File name: hearth_power_up
2  # Author: Pekka Lehtola
3  # Description: Used for creating heart power ups in game.
4
5  from random import randint
6  from heart_ui import Heart_ui
7  from heart_ui import *
8
9  class Heart_power_up(Heart_ui):
10     def __init__(self):
11         super(Heart_power_up, self).__init__()
12         self.surf = pygame.image.load("images/heart_gold.png")
13         self.surf.set_colorkey(GREEN_SCREEN)
14
15
16     def add_power_up(self, hearth_power_up_list):
17
18         self.rect.centerx = randint(300, 1620)
19         self.rect.centery = randint(510, 1000)
20
21         hearth_power_up_list.append(self)
22
23     def show_power_up(self, screen):
24
25         screen.blit(self.surf, self.rect)
26
27     def heart_pickup(self, indian, cowboy, heart):
28
29         if self.rect.colliderect(cowboy.rect):
30             cowboy.health.append(heart)
31             self.rect.x = 2200
32             self.kill()
33
34         if self.rect.colliderect(indian.rect):
35             indian.health.append(heart)
36             self.rect.x = 2200
37             self.kill()
38
```


Ammo power up class:

```
main.py x ammo.py x heart_ui.py x ammo_ui.py x heart_power_up.py x character.py x ui.py x config.py x sup
1  # File name: super_ammo
2  # Author: Pekka Lehtola
3  # Description: super ammo used for ammo power ups.
4
5  from config import *
6  import pygame
7  from random import randint
8  from ammo import Ammo
9
10
11 class Ammo_power_up(pygame.sprite.Sprite):
12
13     def __init__(self):
14         super(Ammo_power_up, self).__init__()
15         self.surf = pygame.image.load("images/super_ammo.png")
16         self.surf.set_colorkey(GREEN_SCREEN)
17         self.rect = self.surf.get_rect()
18
19
20
21     def add_power_up(self, super_ammo_power_up_list):
22
23         self.rect.centerx = randint(300, 1620)
24         self.rect.centery = randint(510, 1000)
25
26         super_ammo_power_up_list.append(self)
27
28     def show_power_up(self, screen):
29
30         screen.blit(self.surf, self.rect)
31
32     def super_pickup(self, indian, cowboy):
33
34         if self.rect.colliderect(cowboy.rect):
35
36             self.rect.x = 2200
37             self.kill()
38
39             cowboy.ammo = []
40             for ammo in range(0, 1):
41
42                 ammo = Ammo("images/bullet.png", "bullet")
43                 ammo.surf = pygame.image.load("images/bullet.png")
44                 ammo.surf = pygame.transform.scale(pygame.image.load("images/bullet.png"),
45                                                     (bullet_x * 10, bullet_y * 10))
46                 ammo.surf.set_colorkey(GREEN_SCREEN)
47                 ammo.rect = ammo.surf.get_rect()
48                 ammo.super = True
49
50                 cowboy.ammo.append(ammo)
51
52         if self.rect.colliderect(indian.rect):
53
54             self.rect.x = 2200
55             self.kill()
56
57             indian.ammo = []
58
59             for ammo in range(0, 1):
60
61                 ammo = Ammo("images/arrow.png", "arrow")
62                 ammo.surf = pygame.image.load("images/arrow.png")
63                 ammo.surf = pygame.transform.scale(pygame.image.load("images/arrow.png"),
64                                                     (arrow_x * 8, arrow_y * 8))
65                 ammo.surf.set_colorkey(GREEN_SCREEN)
66                 ammo.rect = ammo.surf.get_rect()
67                 ammo.super = True
68
69                 indian.ammo.append(ammo)
70
```

Screen captures from the game:



Kun pelaaja kerää super ammo power upin luodin koko ja vahinko minkä tuottaa kasvaa.

Kultaista sydämiä antaa pelaajalle lisää elämää

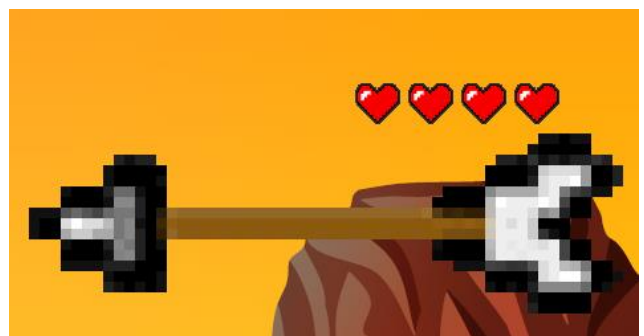
Molemmat power upit ilmestyvät kentälle satunnaisina aikoina



Normaalit amukset:



Super amukset:



Screen capture of git log (showing that you made a commit after every task).

pele98 / Object_Oriented_Programming

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master Object_Oriented_Programming / OOP / Exercise_work / Go to file Add file ...

pele98 Exercise_work implementing power ups 1078f0e 32 minutes ago History

..		
Documents	Exercise_work Week 2 Done	15 days ago
images	Exercise_work implementing power ups	32 minutes ago
TESTING.py	Exercise_work implementing power ups	32 minutes ago
Testing.rar	Exercise_work implementing power ups	32 minutes ago
ammo.py	Exercise_work implementing power ups	32 minutes ago
ammo_ui.py	Exercise_work implementing power ups	32 minutes ago
character.py	Exercise_work implementing power ups	32 minutes ago
config.py	Exercise_work implementing power ups	32 minutes ago
heart_power_up.py	Exercise_work implementing power ups	32 minutes ago
heart_ui.py	Exercise_work implementing power ups	32 minutes ago
main.py	Exercise_work implementing power ups	32 minutes ago
super_ammo.py	Exercise_work implementing power ups	32 minutes ago
ui.py	Exercise_work implementing power ups	32 minutes ago
window.py	Exercise_work initial commit	26 days ago

Self-assessment:

This exercise was easy/difficult/ok/etc. for me because...

Ok. haasteena ilmeni poweruppien ilmestyminen satunnaisen ajan kuluttua ja tietylle alueelle.

Haaste oli myös saada ne toimimaan oikien.

Doing this exercise, I learned...

Pygame.scale metodin käyttöä

I am still wondering...

-

I understood/did not understand that... ; I did/did not know that... ; I did/did not manage to do...

Power upit toimivat vielä hiukan bugisesti ja tosiaan aika ei riittänyt koodin viimeistelyyn. Korjaan Asian viimeisessä palautuksessa.